The AMD Opteron Processor for Multiprocessor Servers

Representing AMD's entry into 64-bit computing, Opteron combines the backwards compatibility of the x86-64 architecture with a DDR memory controller and HyperTransport links to deliver server-class performance. These features also make Opteron a flexible, modular, and easily connectable component for various multiprocessor configurations.

Chetana N. Keltcher
Kevin J. McGrath
Ardsher Ahmed
Pat Conway
Advanced Micro Devices

Advanced Micro Devices' Opteron processor is a 64-bit, x86-based microprocessor with an on-chip double-data-rate (DDR) memory controller and three HyperTransport links for glueless multiprocessing. AMD based Opteron on the x86-64 architecture,[1] AMD's backward-compatible extension of the industry standard x86 architecture. Thus, Opteron provides seamless, high-performance support for both the existing 32-bit x86 code base and new 64-bit software. Its core is an out-of-order, superscalar processor supported by large on-chip level-1 (L1) and level-2 (L2) caches. Figure 1 shows a block diagram. A key feature is the on-chip, integrated DDR memory controller, which leads to low memory latency and provides high-performance support to integer, floating-point, and multimedia instructions. The HyperTransport links let Opteron connect to neighboring Opteron processors and other HyperTransport devices without additional support chips.[2]

Why 64 bit?

The need for a 64-bit architecture arises from applications—such as databases, computer-aided design (CAD) tools, scientific/engineering problems, and high-performance servers—that address large amounts of virtual and physical memory. For example, the Hammer design project, which includes Opteron, ran most of its design simulations on a compute farm of 3,000 Athlons running Linux. However, for some large problem sets, our CAD tools required greater than 4 Gbytes of memory and had to run on far more expensive platforms that supported larger addresses. The Hammer project could have benefited from the existence of low-cost 64-bit computing.

However, there exists a huge software code base built around the legacy 32-bit x86 instruction set. Many x86 workstation and server users now face the dilemma of how to transition to 64-bit systems and yet maintain compatibility with their existing code base. The AMD x86-64 architecture adds 64-bit addressing and expands register resources to support higher performance for recompiled 64-bit programs. At the same time, it also supports legacy 32-bit applications and operating systems without modification or recompilation. It thus supports both the vast body of existing software as well

Published by the IEEE Computer Society

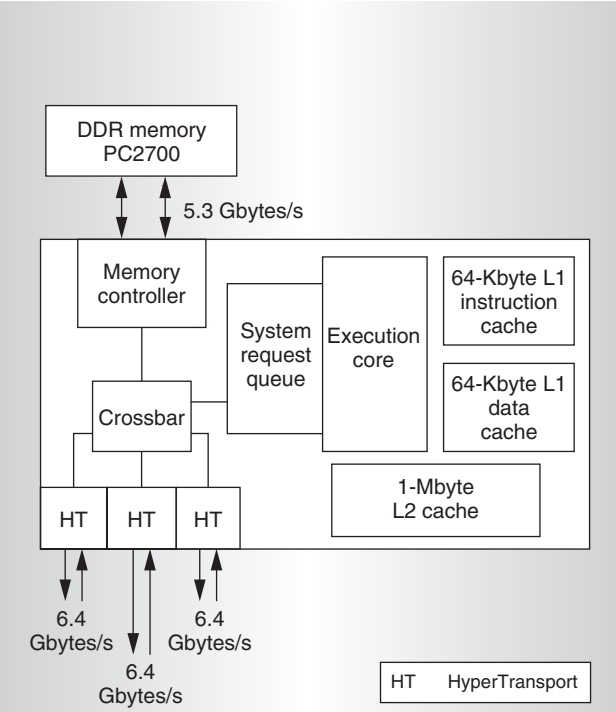

Figure 1. Opteron architecture.

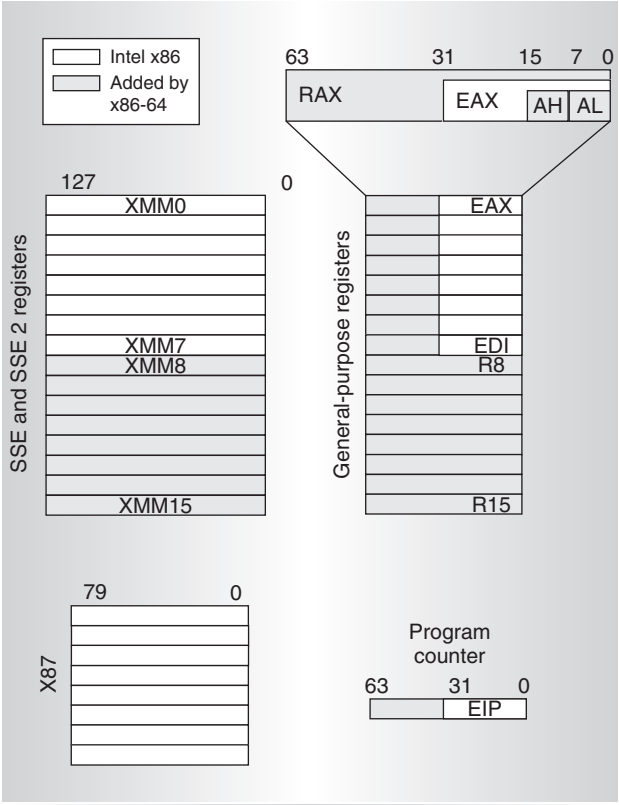

Figure 2. Programmer's model for the x86-64 architecture.

as new 64-bit software.

The AMD x86-64 architecture supports 64-bit virtual and 52-bit physical addresses, although a given implementation might support less. The Opteron, for example, supports 48-bit virtual and 40-bit physical addresses. The x86-64 architecture extends all x86 integer arithmetic and logical integer instructions to 64 bits, including 64 × 64-bit multiply with a 128-bit result.

The x86-64 architecture doubles the number of integer general-purpose registers (GPRs) and streaming SIMD extension (SSE) registers from 8 to 16 of each. It also extends the GPRs from 32 to 64 bits. The 64-bit GPR extensions overlay the existing 32-bit registers in the same way as they had in the x86 architecture's previous transition, from 16 to 32 bits.

Figure 2 depicts the programmer's view of the x86-64 registers, showing the legacy x86 registers and the x86-64 register extensions. Programs address the new registers via a new instruction prefix byte called REX (register extension).

The extension to the x86 register set is straightforward and much needed. Current x86 software is highly optimized to work within the small number of legacy registers. Adding eight more registers satisfied the register allocation needs of more than 80 percent of functions appearing in typical application code, as Figure 3 shows.

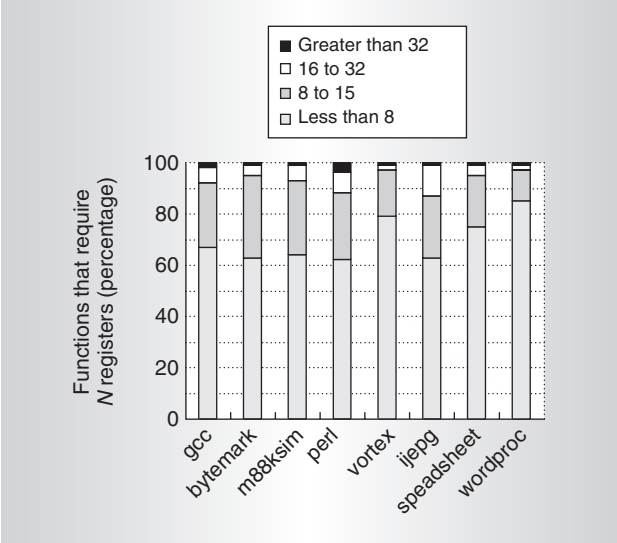

Figure 3. Register usage in typical applications.

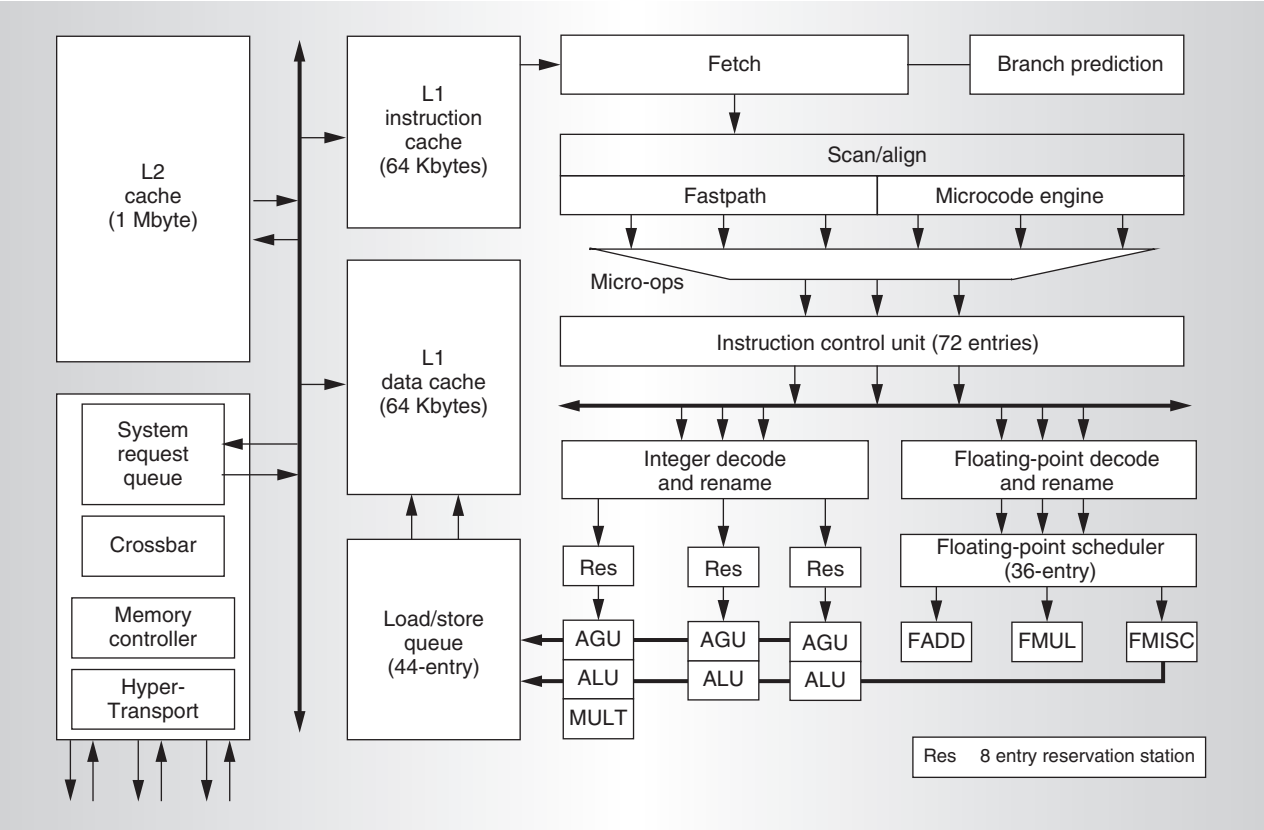

Figure 4. Processor core block diagram.

Core microarchitecture

Figure 4 is a block diagram of the Opteron core, an aggressive, out-of-order, three-way superscalar processor. It reorders as many as 72 micro-ops, and fetches and decodes three x86-64 instructions each cycle from the instruction cache. Like its predecessor, the AMD Athlon,[3] Opteron turns variable-length x86-64 instructions into fixed-length micro-ops and dispatches them to two independent schedulers: one for integer, and one for floating-point and multimedia (MMX, 3Dnow, SSE, and SSE 2) operations. In addition, it sends load and store micro-ops to the load/store unit. The schedulers and the load/store unit can dispatch 11 micro-ops each cycle to the following execution resources:

- three integer execution units,
- three address generation units,
- three floating point and multimedia units, and
- two loads/stores to the data cache.

Pipeline

Opteron's pipeline, shown in Figure 5, is long enough for high frequency and short enough for good IPC (instructions per cycle). This pipeline is fully integrated from instruction fetch through DRAM access. It takes seven cycles for fetch and decode; excellent branch prediction covers this latency. The execute pipeline is typically 12 stages for integer and 17 stages for floating-point. The data cache access occurs in stage 11 for loads; the result returns to the execution units in stage 12.

In case of an L1 cache miss, the pipeline accesses the L2 cache and (in parallel) a request goes to the system request queue. The pipeline cancels the system request on an L2 cache hit. The memory controller schedules the request to DRAM.

The stages in the DRAM pipeline run at the same frequency as the core. Once data is returned from DRAM, it takes several stages to route data across the chip to the L1 cache and to perform error code correction. The

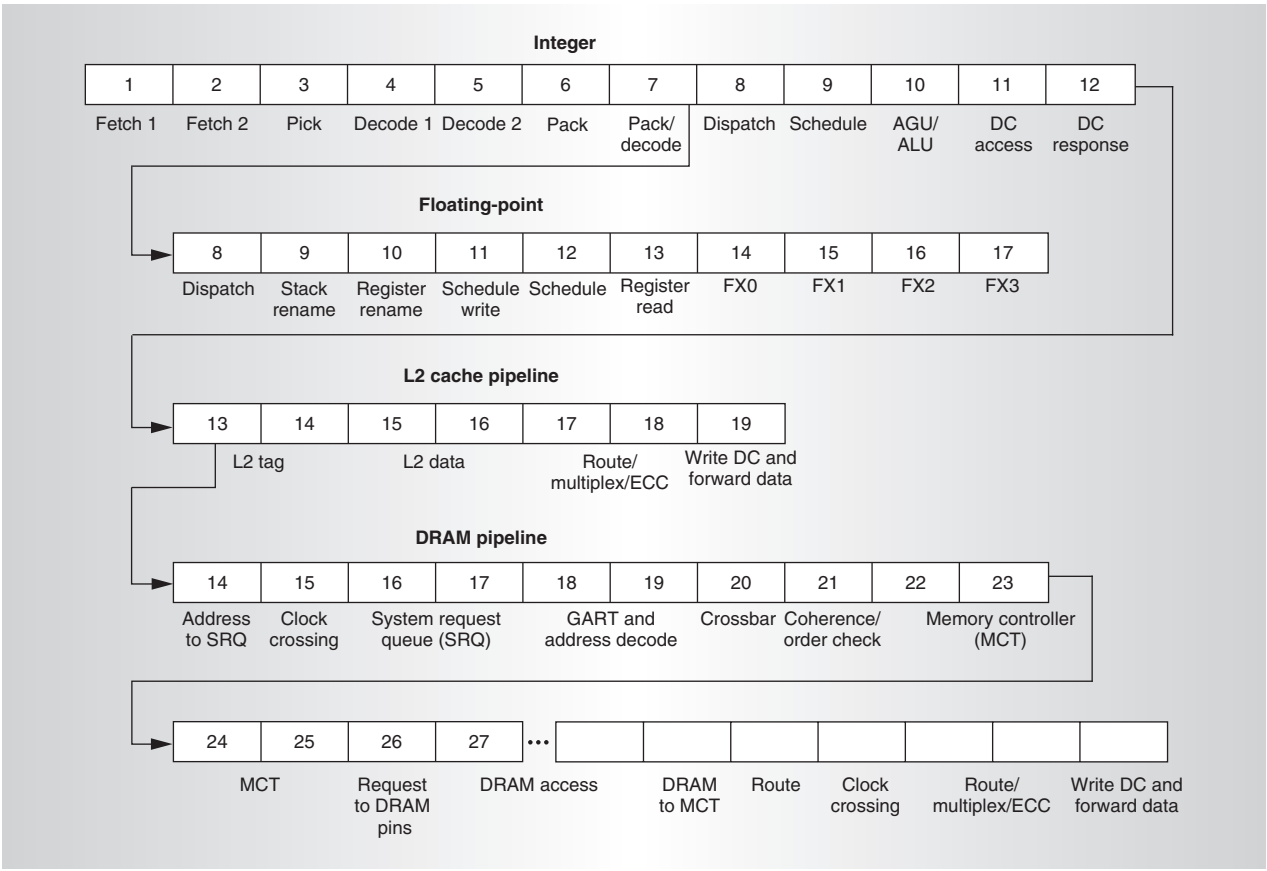

Figure 5. Opteron pipeline.

processor forwards both the L2 cache and system data, critical-word first, to the waiting load while updating the L1 cache.

Caches

Opteron has separate L1 data and instruction caches, each 64 Kbytes. They are two-way set-associative, linearly indexed, and physically tagged with a cache line size of 64 bytes. We banked them for speed with each way consisting of eight 4-Kbyte banks. Backing these L1 caches is a large 1-Mbyte L2 cache, whose data is mutually exclusive with respect to the data in the L1 caches. The L2 cache is 16-way set-associative and uses a pseudo-least-recently-used (LRU) replacement policy, grouping two ways into a sector and managing the sectors via an LRU criterion. This mechanism uses only half the number of LRU bits of a traditional LRU scheme, with similar performance. We used the standard MOESI (modified, owner, exclusive, shared, invalid) protocol for cache coherence.

The instruction and data caches each have independent L1 and L2 translation look-aside buffers (TLB). The L1 TLB is fully associative and stores thirty-two 4-Kbyte page translations and eight 2-Mbyte/4-Mbyte page translations. The L2 TLB is four-way set-associative with 512 4-Kbyte entries. In a classical x86 TLB scheme, a change to the page directory base flushes the TLB. Opteron has a hardware flush filter that blocks unnecessary TLB flushes and flushes the TLBs only after it detects a modification of a paging data structure. Filtering TLB flushes can be a significant advantage in multiprocessor workloads because it lets multiple processes share the TLB.

Instruction fetch and decode

The instruction fetch unit, with the help of branch prediction, attempts to supply 16 instruction bytes each cycle to the scan/align unit, which scans this data and aligns it on instruction boundaries. This task is compli-

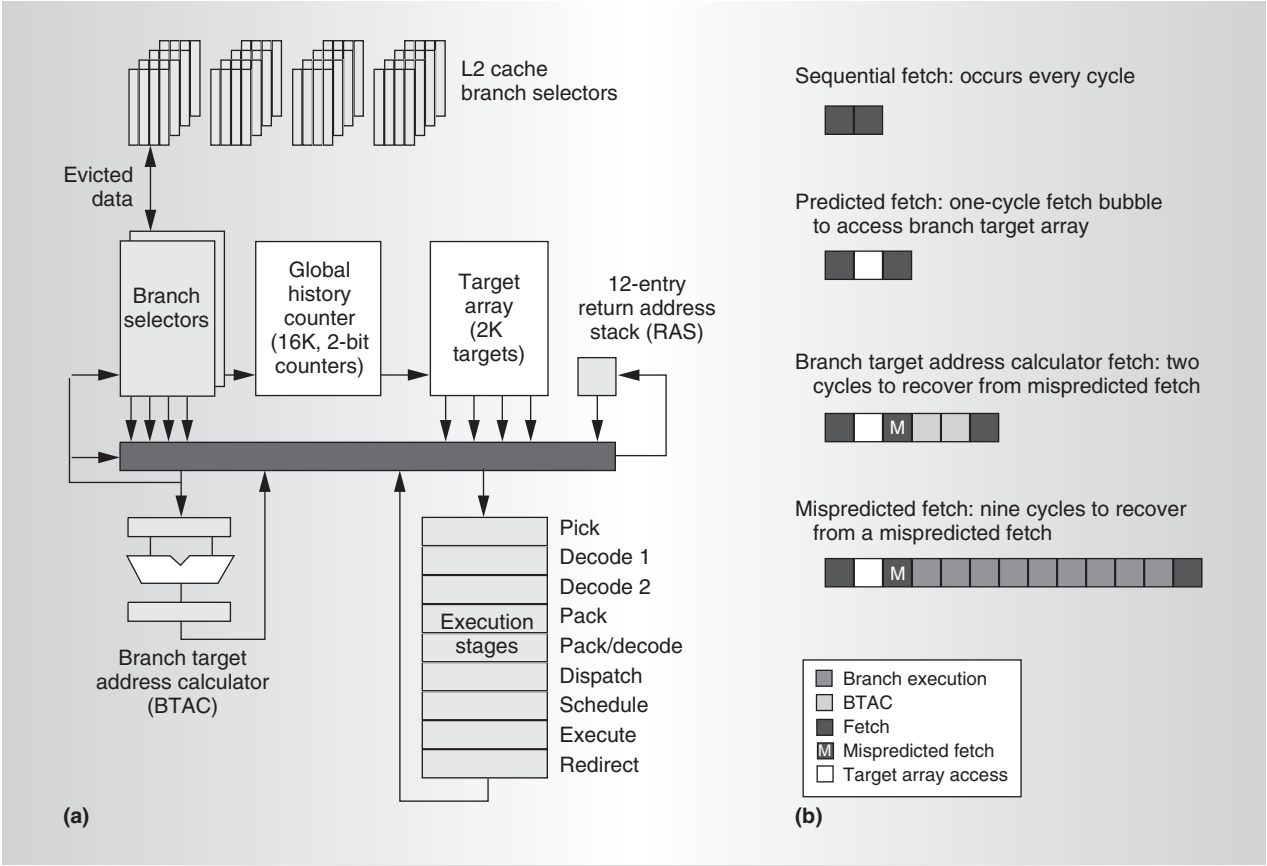

Figure 6. Branch predictor block diagram (a) and diagram of branch prediction schemes (b). Each set of boxes in the diagram represents a sequence of cycles that occur for each scheme.

cated because x86 instructions can vary in length from 1 to 15 bytes. To simplify the subsequent scan process, the instruction fetch unit marks instruction boundaries as it writes the line into the instruction cache.

The processor decodes variable-length instructions into internal fixed-length micro-ops using either the hardware decoders (called "fastpath" decoders) or the microcode engine. Fastpath decoders can issue as many as three x86 instructions per cycle. Most common x86 instructions that decode to one or two micro-ops per instruction use the fastpath decoders. On the other hand, microcode can issue only one x86 instruction per cycle. For better performance, Opteron has more fastpath decoding resources than its predecessor, Athlon. For example, packed SSE instructions were microcoded in Athlon, but they are fastpath in Opteron. This change resulted in 8 percent fewer microcoded instructions for SPECint benchmarks and 28 percent fewer for SPECfp.

Branch prediction

AMD significantly improved Opteron's branch prediction over that of Athlon. As Figure 6 shows, Opteron uses a combination of branch prediction schemes.[4] The branch selector array selects between static prediction and the history table.

The global history table uses 2-bit saturating up/down counters. The branch target array holds branch target addresses and has backup from a 2-cycle branch target address calculator to accelerate address calculations. The return-address stack optimizes CALL/RETURN pairs by storing the return address of each CALL and supplying it for use by the corresponding RETURN instruction. The mispredicted branch penalty is 11 cycles. When a line is evicted from the instruction

cache, it saves the branch prediction information and the end bits in the L2 cache error correction code field, because parity protection is sufficient for read-only instruction data. Thus, the fetch unit does not have to recreate branch history information when it brings a line back into the instruction cache from the L2 cache. This unique and simple trick improved Opteron's branch prediction accuracy on various benchmarks by 5 to 10 percent over the Athlon.

Integer and floating-point execution units

The decoders dispatch three micro-ops each cycle to the instruction control unit, which has a 72-entry reorder buffer and controls the retirement of all instructions. In parallel, the decoders issue micro-ops to the integer reservation stations and the floating-point scheduler. In turn, these units issue micro-ops to their respective execution units once the instructions' operands are available. Opteron has three units for each of the following functions: integer execution, address generation, and floating-point execution. The integer units have a full 64-bit data path, and the majority of the micro-ops (32- and 64-bit add, subtract, rotate, shift, logical, and so on) are single cycle. The hardware multiplier takes 3 cycles for a 32-bit multiply and 5 cycles for a 64-bit multiply.

The floating-point data path is full 80-bit extended precision. All floating-point units are fully pipelined with a throughput of one instruction per cycle for most operations. Simple operations like compare and absolute as well as most MMX operations take 2 cycles. MMX multiply and sum-of-average instructions take 3 cycles. Single- and double-precision add and multiply instructions take 4 cycles, and the latency of divide and square root instructions depends on precision. The divide takes 16, 20, or 24 cycles for single, double, or extended-precision data; the square root takes either 19, 27, or 35 cycles.

Load store unit

In addition to the integer and floating-point units, the decoders issue load/store micro-ops to the load/store unit, which manages load and store accesses to the data cache and system memory. Each cycle, two 64-bit loads or stores can access the data cache as long as they are to different banks. Load-to-use latency is 3 cycles. Loads require an extra cycle if they have a nonzero segment base, are misaligned, or create bank conflicts. Although loads can return results to the execution units out of order, stores cannot commit data until they retire. Stores that have not yet committed forward their data to dependent loads. A hardware prefetcher[5] recognizes a pattern of cache misses to consecutive lines (ascending or descending) and prefetches the next cache line into the L2 cache. The L2 cache can handle 10 outstanding requests for cache miss, state change, and TLB miss—eight from the data cache and two from the instruction cache.

Memory controller and HyperTransport

The on-chip memory controller provides a dual-channel 128-bit wide interface to a 333-MHz DDR memory composed of PC2700 dual, in-line memory modules (DIMMs). The controller's peak memory bandwidth is 5.3 Gbytes/s. It supports eight registered or unbuffered DIMMs for a memory capacity of 16 Gbytes using 2-Gbyte DIMMs. It is on a separate clock grid, but runs at the same frequency as the core. Thus, memory latency scales down with frequency improvements in the processor core. In low-power modes, the processor can turn off the core clock while keeping the memory controller's clock running to permit access to memory by graphics and other devices. Historically, the memory controller has been relegated to a separate chip called the Northbridge, as illustrated in Figures 7a and 7b. The integrated memory controller significantly reduces Opteron's memory latency and improves performance by 20 to 25 percent over Athlon.

Other on-chip Northbridge functionality includes the processing of I/O direct-mapped access, memory-mapped I/O, peripheral component interconnect (PCI) configuration requests, transaction ordering, and cache coherence. A broadcast cache coherence protocol maintains memory coherence by avoiding the serialization delay of directory-based systems and allowing the snooping of the processor caches concurrently with DRAM access. The snoop throughput scales with processor frequency and HyperTransport link speed.

Among Opteron's unique features are the

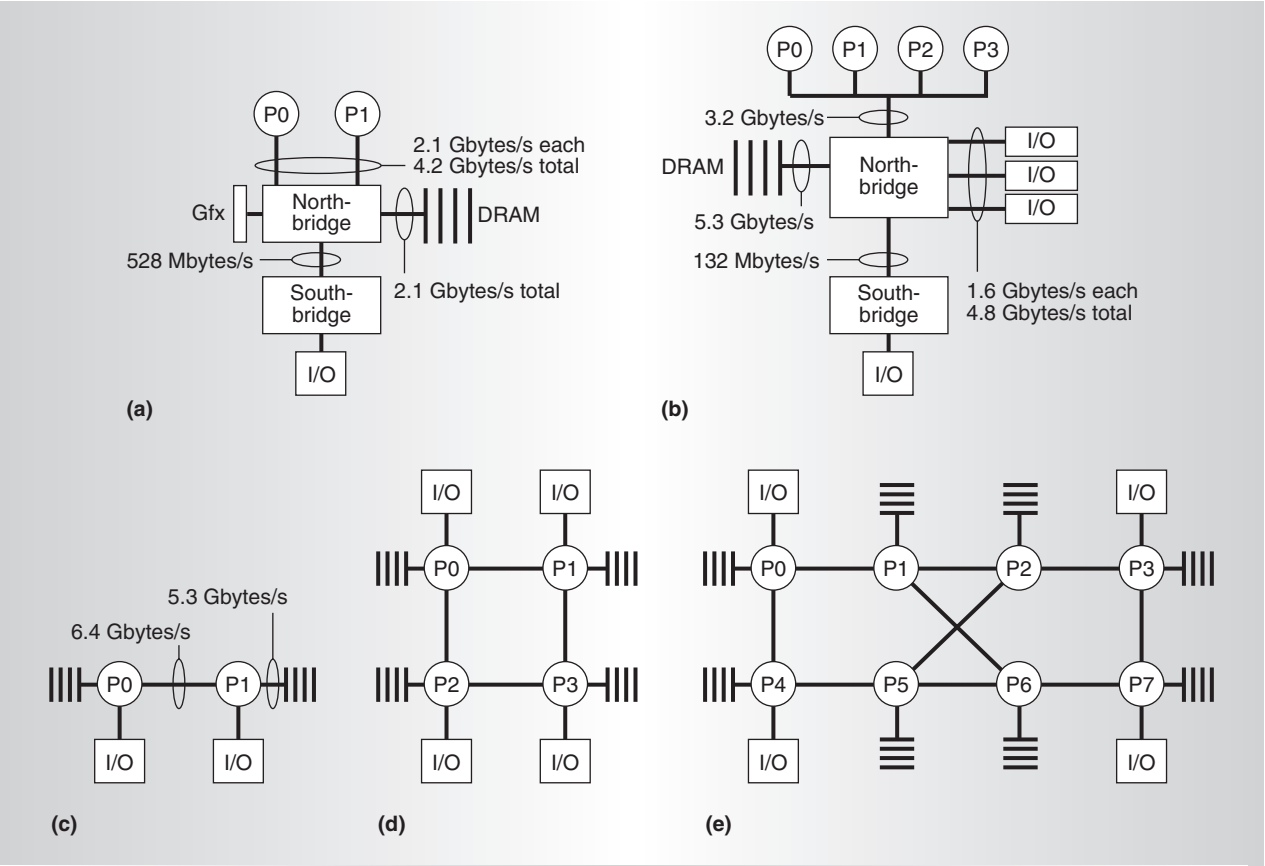

Figure 7. Multiprocessor server architectures: two-way Athlon (a), four-way Intel Xeon (b), and two-way (c), four-way (d), and eight-way (e) Opteron.

three integrated HyperTransport links. Each link is 16 bits wide (3.2 Gbytes/s per direction) and is configurable as either coherent HyperTransport (cHT) or noncoherent HyperTransport (HT). The cHT protocol is for connecting processors, and the simpler HT protocol is for attaching I/O. The point-to-point HyperTransport links support flexible, configurable, and scalable I/O topologies using chains of I/O "tunnel" and "cave" devices. The programming model for the configuration space is identical to that for the PCI bus specification, version 2.2.[6] The HyperTransport links are transparent to any OS that supports PCI.

Reliability features

Reliability is very important in enterprise-class machines, and Opteron has robust reliability features. Error code correction (ECC) or parity checking protects all large arrays. The L1 data cache, L2 cache, and DRAM are ECC protected. Eight ECC check bits are stored for each 64 bits of data, allowing the correction of single-bit errors and the detection of double-bit errors (SECDED, which stands for single error correction double error detection). DRAM has support for chipkill ECC, which allows for the correction of 4-bit errors in the same nibble. The processor accomplishes this by using 16 ECC check bits on 128 bits of data. Thus, chipkill ECC can correct errors affecting an entire x4 data width DRAM chip. In addition, the L1 data cache, L2 cache tags, and DRAM implement hardware scrubbers that steal idle cycles and clean up single-bit ECC errors. This clean-up is particularly important for the L1 data cache, which detects but does not correct errors on the fly. Parity checking protects the instruction cache data, and L1 and L2 TLBs. ECC and parity-checking errors are reported via a machine check architecture that reports failures with sufficient information to diagnose the error. The

highly integrated Opteron design results in lower overall chip count, which also improves system reliability.

Processor core performance

An Opteron processor operating at 2.0 GHz using registered 333-MHz DDR memory has estimated scores of 1,202 for SPECint2000 and 1,170 for SPECfp2000.

Glueless multiprocessing

Here, we illustrate the multiprocessing design space for Opteron system architectures and present microbenchmark latency and bandwidth results for shared-memory accesses from processors and I/O nodes. These microbenchmark results show that coherent memory bandwidth scales. Memory latencies are such that the programmer has a near-flat view of the memory.

Figure 7 shows the architectures of traditional two- and four-way x86 servers, in which all the processors in the system share a single, external memory controller. In contrast, an integrated memory controller provides a 128-bit 333-MHz DDR interface at each node in an Opteron system. A multiprocessor configuration can use memory on a node contiguously or interleave pages from one node's memory with those from other nodes in the system. Three on-chip HyperTransport links enable streaming server I/O and high-bandwidth, glueless two-, four-, and eight-way multiprocessor systems, as shown in Figures 7c, 7d, and 7e.

Multiprocessor performance analysis

To evaluate the performance of our multiprocessor systems, we ran the following register-transfer-level microbenchmarks, measuring latencies and bandwidths on one, two, and four processor systems:

- *Local memory access.* In this microbenchmark, each processor generates a series of read-memory transactions to addresses in local memory only. We measured the isolated read latency as well as bandwidth under saturated load. Figure 8a shows this communication pattern.
- *Xfire memory access.* In the Xfire (crossfire) microbenchmark, all processors read data from all nodes in an interleaved manner, such that the generated system load corresponds to all-to-all communication. We measure the isolated read latency as well as bandwidth under saturated load; Figure 8b shows this communication pattern.

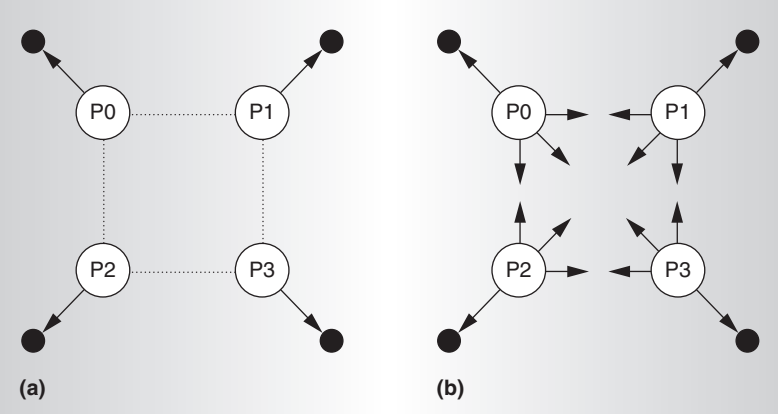

Figure 8. Local memory access (a) and Xfire memory access (b) benchmarks measure system performance under varying communication patterns.

The number of DIMMs and the memory capacity double when we double the number of processors. There are three I/O HyperTransport links for one processor and four I/O HyperTransport links for two- and four-processor configurations, each link providing 6.4 Gbytes/s of bandwidth. The processors are gluelessly interconnected with coherent HT links for the efficient topologies shown in Figures 7c, 7d, and 7e. For topologies 7c and 7d, the average distance is less than or equal to 1 hop between nodes, whereas for 7e, it is 1.75 hops. Table 1 summarizes these results. The memory-read bandwidth in a four-processor system accessing data in local memory is up to 15.59 Gbytes/s; bandwidth is 11.23 Gbytes/s when the system interleaves memory across all the nodes.

When you bisect a network (cut it into two parts with an equal number of processors) such that the number of links cut is a minimum, it is called a min cut. The bandwidth across a min cut is the *bisection bandwidth*, a measure of available bandwidth in the system for interprocessor communications.

The graph in Figure 9 shows that the read-memory bandwidth scales from one- to four-processor systems. The average unloaded memory latency with open DRAM pages for

Table 1. Microbenchmark results.

System parameters	One processor	Two processors	Four processors
No. of DIMMs	8	16	32
Total main-memory using 2-Gbyte DIMMs (Gbytes)	16	32	64
No. of HyperTransport I/O links	3	4	4
Bisection bandwidth (Gbytes/s)	NA	6.4	12.8
Diameter (no. of hops)	NA	1	2
Average distance (no. of hops)	NA	0.5	1
Local memory read bandwidth (Gbytes/s)	5.3	10.67	15.59
Local bandwidth per processor (Gbytes/s)	5.3	5.3	3.9
Xfire memory read bandwidth (Gbytes/s)	5.3	7.06	11.23
Xfire bandwidth per processor (Gbytes/s)	5.3	3.53	2.8

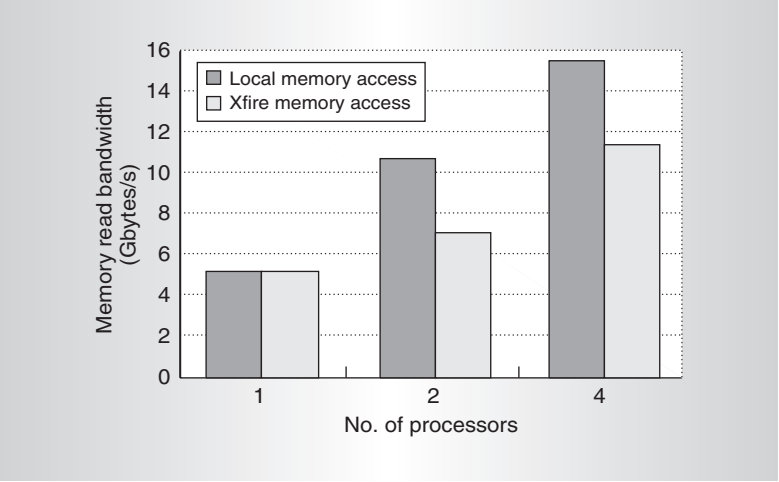

Figure 9. Memory bandwidth scalability.

one-, two-, and four-processor systems is less than 50 ns, less than 70 ns, and less than 110 ns. The latency difference between remote and local memory in a given system is comparable to the difference between a DRAM page hit and a DRAM page conflict. The latency under load increases slowly because of the availability of excess interconnect bandwidth. Latency shrinks quickly with increasing CPU clock speed and HyperTransport link speed. The aggregate memory-read bandwidths for one-, two-, and four-processor systems are 5.3 Gbytes/s, 7 Gbytes/s, and 11.23 Gbytes/s, where the data resides uniformly across the nodes.

Scaling beyond eight-way multiprocessing

We designed Opteron for use as a building block for highly scalable and reliable enterprise servers.[7,8] It is designed to scale beyond eight ways by combining the processor with an external HyperTransport switch, as illustrated in Figure 10. The average distance between any pair of nodes in this topology is 2.7 hops, and the maximum distance between any pair of nodes (the network diameter) is 5 hops. The switch contains a snoop filter, which tracks the state of cached lines in the system. Requests access the snoop filter to locate the latest copy of a line in the system. The switch can also contain a large cache to hold data, which is remote to a quad. Figure 10 shows the quad configuration, a group of four processors, each with an independent I/O and local memory. The four processors share two sets of remote caches and snoop filters; they connect to the rest of the system through two coherent switches (SW0 and SW1) and six HyperTransport links.

Both the snoop filter and the remote cache reduce traffic among switches and help bandwidth scale with processor count.

The integrated DDR memory controller, glueless multiprocessing capability, high-bandwidth HyperTransport links, 64-bit computing, and high-reliability features combined with a core that provides leading-edge performance makes the AMD Opteron a great processor for server applications. We plan future microarchitectural improvements such as support for next-generation DDR technology, enhancements to the cache and memory subsystem, and support for larger linear

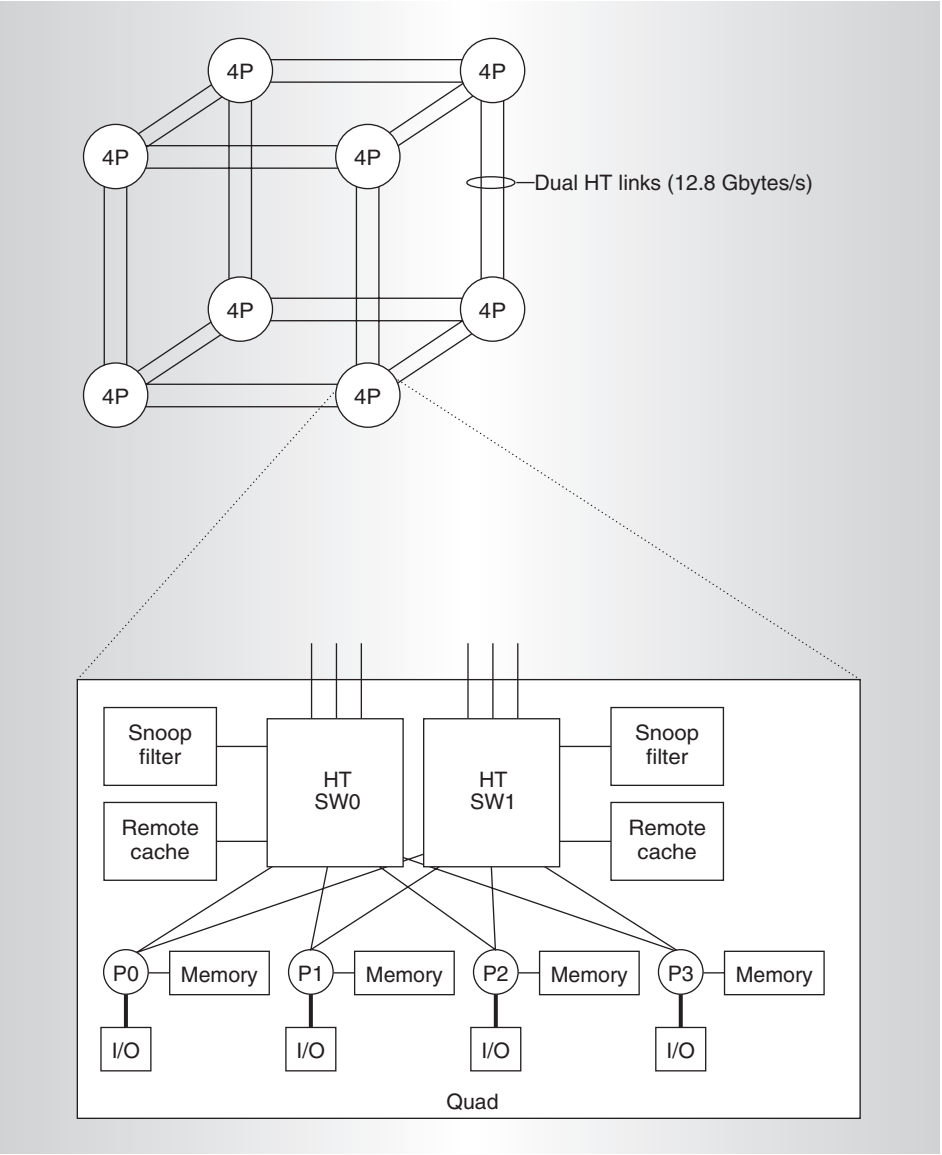

Figure 10. Scaling beyond eight-way multiprocessing using an external switch. This configuration provides 256 DIMMs, 512-Gbytes of total memory built from 256-Mbyte DRAMs, 32 HT links, and a bisection bandwidth of 51.2 Gbytes/s.

and physical addresses. In addition to servers, the core architecture described here serves as a foundation for desktops, laptops, and workstations.

AMD targets Opteron processors for fabrication in a 130-nm silicon-on-insulator process with nine layers of copper interconnect, a process available at AMD's fab in Dresden, Germany. In the future, AMD plans to move to a 90-nm process. Although the design team has focused on eliminating unnecessary power dissipation without sacrificing performance by extensive use of clock gating, we plan further work to lower power requirements. MICRO

Acknowledgments

We thank all the members of the Opteron architecture team, including Fred Weber, Bill Hughes, Ramsey Haddad, Dave Christie, Scott White, Gerald Zuraski, Phil Madrid, Kelvin Goveas, Ashraf Ahmed, Michael Clark, Hongwen Gao, Bruce Holloway, Richard Klass, Dave Kroesche, and Tim Wood.

References

1. *AMD x86-64 Architecture Manuals*, http://www.amd.com.
2. *HyperTransport I/O Link Specification*, http://www.hypertransport.org/tech_specifications.html.
3. D. Meyer, "The AMD-K7 Microprocessor: A First Look," *Microprocessor Forum*, 1998; http://www.mdronline.com/mpf/.
4. S. McFarling, "Combining Branch Predictors," WRL Technical Note TN-36, Compaq Western Research Laboratory, 1993.
5. T-F. Chen and J-L. Baer, "Effective Hardware-Based Data Prefetching for High-Performance Processors," *IEEE Trans. Computers*, vol. 44, no. 5, May 1995, pp. 609-623.
6. *PCI Local Bus Specification*, revision 2.2, 18 December 1998; http://www.pcisig.com/specifications.
7. F. Aono and M. Kimura, "The AzusA 16-way Itanium server," *IEEE Micro*, vol. 20, no. 5, Sept.-Oct. 2000, pp. 54-60.
8. D. Watts et al., "Technical Description," *IBM xSeries 440 Planning and Installation Guide*, Oct. 2002; http://www.redbooks.ibm.com/redbooks/SG246196.html.

Chetana N. Keltcher is a member of the technical staff at Advanced Micro Devices. She worked on the K6 and Athlon microprocessors and is now part of the Opteron architecture team where she is responsible for the load-store and data cache design. Keltcher has a BE in computer science from Bangalore University, India, and an MS and PhD in computer science from the Pennsylvania State University. She is a member of the IEEE.

Kevin J. McGrath is a fellow at AMD, California Microprocessor Division. He is the architect of AMD's 64-bit extensions and currently manages the Opteron architecture and RTL team. His work experience includes 20 years in CPU design and verification, first for Hewlett-Packard and later for ELXSI, eventually working on the microarchitecture of the Nx586, K6, and Athlon processors, leading the microcode team for those projects. McGrath has a BS in engineering technology from California Polytechnic University, San Luis Obispo.

Ardsher Ahmed was a senior member of the technical staff, design engineer, at AMD. His research interests include designing high-performance computer architectures; multiprocessor systems; server development; supercomputing and massively parallel processing; high-bandwidth low-latency interconnection network design; performance evaluation; and microbenchmarking. Ahmed has a BS in electrical engineering from NED University of Engineering and Technology, Karachi, Pakistan; an MS in electrical engineering from the California Institute of Technology, Pasadena; and a PhD in computer engineering from Binghamton University, New York. He is a member of the IEEE.

Pat Conway is a senior member of technical staff in the Computational Products Group at AMD where he is responsible for developing scalable, high-performance server architectures. His work experience includes the design and development of server hardware, cache coherence protocols, user-level message passing, and clustering technology. Conway has an MEngSc from the University College Cork, Ireland, and an MBA from Golden Gate University. He is a member of the IEEE.

For more information on AMD Opteron, see the Web site at http://www.amd.com/opteron/.